LISTING OF CLAIMS

The listing of claims provided below replaces all prior versions, and listings, of claims in the application.

1. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) computation on a message, comprising:

a core having a first execution unit and a second execution unit, wherein the first execution unit is defined to perform a schedule computation on a data block of the message, ~~capable of processing a message and producing a partial result passed to the second execution unit~~, the first execution unit defined to communicate a partial result of the schedule computation on the data block to the second execution unit when the partial result becomes available and prior to completion of the schedule computation on the data block, wherein the second execution unit is defined to perform a compression function on the partial result received from the first execution unit ~~the partial result capable of being processed by the second execution unit~~ in parallel with ~~the processing of the message by~~ the first execution unit continuing the schedule computation on the data block.

2. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) of claim 1, wherein the first execution unit is a single instruction multiple data (SIMD) execution unit.

3. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) of claim 1, wherein the second execution unit is an integer execution unit.

4. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) of claim 1, wherein the message is a parsed padded message.

5. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) of claim 4, wherein the parsed padded message includes an original message and a plurality of pad bits, the original message being a plurality of bits.

6. (Currently Amended) A processor for ~~capable of~~ executing a secure hash algorithm (SHA) of claim 1, wherein the partial result includes a group of bits ~~capable of being~~ represented as [[~~by~~]] a hexadecimal value.

7. (Currently Amended) A processor for cryptographic computation, comprising:

a first execution unit defined to perform ~~capable of performing~~ a message schedule computation on a data block and produce ~~producing~~ a partial result of the schedule computation on the data block prior to completion of the schedule computation on the data block, wherein the partial result includes a group of bits capable of being represented by a hexadecimal value; and

a second execution unit defined to perform ~~capable of performing~~ a compression function on [[~~using~~]] the partial result while the first execution unit continues performing the message schedule computation on the data block~~, wherein the second execution unit is capable of operating in parallel with the first execution unit~~.

8. (Currently Amended) A processor for cryptographic computation of claim 7, wherein the first execution unit is defined to receive ~~receives~~ a plurality of blocks, the plurality of blocks including an original message and a plurality of pad bits.

9. (Currently Amended) A processor for cryptographic computation of claim 8, wherein the first execution unit is defined to perform a rotation operation on the plurality of blocks as part of the message schedule computation ~~includes a rotation operation capable of rotating the plurality of blocks~~.

10-11. (Cancelled)

12. (Currently Amended) A method, comprising:

receiving a message; and

performing a cryptographic computation on the message, the cryptographic computation including ~~being capable of,~~

~~performing~~ a hash computation including ~~such that the cryptographic computation includes operations for~~,

performing a message schedule computation on a block of data using a first execution unit ~~with a block of data~~, whereby a partial result of the message schedule computation is generated prior to completion of the message schedule computation,

communicating the ~~producing a~~ partial result from the first execution unit to a second execution unit while the message schedule computation on the block of data continues using the first execution unit, and performing a compression function on the partial result using the [[~~a~~]] second execution unit while the message schedule computation on the block of data continues using the first execution unit ~~with the partial result in parallel with the message schedule computation~~.

13. (Currently Amended) A method of claim 12, wherein the cryptographic computation includes ~~is further capable of performing~~ a preprocessing operation including,

padding the message to generate a padded version of the message;

parsing the padded version of the message; and

setting initial hash values to be used in the hash computation.

14. (Cancelled)

15. (Original) A method of claim 12, wherein performing the message schedule computation further includes assigning rotated bits in the block of data to the partial result.

16. (Cancelled)

17. (Currently Amended) A method for a one-way cryptographic hash computation, comprising:

operating a first execution unit to perform a message schedule computation on a data block to produce ~~processing a block in a first execution unit and producing~~ a partial result of the message schedule computation on the data block;

sending the partial result from the first execution unit to a second execution unit while the first execution unit continues to operate to perform the message schedule computation on the block of data; and

operating a second execution unit to perform a compression function on ~~processing~~ the partial result while the first execution unit continues performing the message schedule computation on the data block ~~in parallel with the first execution unit~~.

18. (Currently Amended) A method for a one-way cryptographic hash computation of claim 17, wherein operating the first execution unit to perform the message schedule computation ~~processing the block further~~ includes rotating bits in the data block~~, the bits in the block capable of being represented as a hexadecimal value~~.

19. (Currently Amended) A method for a one-way cryptographic hash computation of claim 17, wherein operating the second execution unit to perform the compression function ~~processing the partial result further~~ includes rotating bits in the partial result~~, the bits in the block capable of being represented as a hexadecimal value~~.

20-27. (Cancelled)